Attorney's Docket No.: 9209-10 PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re: Tabares et al.
Serial No.: 09/992,155
Filed: November 5, 2001
For: METHODS, SYSTEMS AND COMPUTER PROGRAM PRODUCTS FOR INSTANTIATING A DEVICE DRIVER FOR COMMUNICATION WITH A DEVICE BY DYNAMICALLY ASSOCIATING THE DEVICE DRIVER AT RUN-TIME WITH A DEVICE-SPECIFIC AND/OR SERVICE-SPECIFIC SOFTWARE COMPONENT

Examiner: Diem K. Cao
Group Art Unit: 2194
Confirmation No.: 5291

October 30, 2007

Mail Stop Appeal Brief-Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

APPELLANTS' SUPPLEMENTAL

BRIEF ON APPEAL UNDER 37 C.F.R. §41.37

Sir:

This Appeal Brief is filed pursuant to the "Notice of Appeal to the Board of Patent Appeals and Interferences" filed October 30, 2007 requesting reinstatement of the appeal and the decision to reopen prosecution as set forth in the Office Action dated July 30, 2007 (hereinafter "Office Action").

Real Party In Interest

The real party in interest is assignee Trendium, Inc., Sunrise, Florida.

Related Appeals and Interferences

Appellants are aware of no appeals or interferences that would be affected by the present appeal.

Status of Claims

Appellants appeal the rejection of Claims 1 – 15, 20 – 34, and 39 – 53 as set forth in the Office Action, which as of the filing date of this Brief remain under consideration. Claims 16 – 19, 35 – 38, and 54 - 57 have been canceled. Appellants submit that the claims involved in the appeal are independent Claims 1, 12, 20, 31, 39, and 50 and the rejected dependent Claims 2 – 11, 13 – 15, 21 – 30, 32 – 34, 40 – 49, and 51 – 53 as a reversal of the rejection of independent Claims 1, 12, 20, 31, 39, and 50 is requested in the present appeal and a reversal of the rejection of dependent Claims 2 – 11, 13 – 15, 21 – 30, 32 – 34, 40 – 49, and 51 – 53 is also requested based on the reversal of the rejection of the independent claims. Accordingly, the pending claims as included in Appellants' response to the Office Action of May 16, 2006 are attached hereto as Appendix A.

Status of Amendments

No responses after final rejection have been filed in the present case.

Summary of Claimed Subject Matter

Independent Claim 1 is directed to a computer implemented method of instantiating a device driver in which a first software component is dynamically associated with the device driver at run-time. The first software component contains information that facilitates communication with devices of a specific device type. (Specification, page 10, lines 13 – 18; FIG. 6).

Independent Claim 12 is directed to a computer implemented method of collecting data from a device. A request to collect data is received from the device. (Specification, page 11, lines 15 – 16; FIG. 9, block 132). A software component is dynamically associated with a device driver at run-time. The software component contains information that facilitates communication with the device. (Specification, page 11, lines 21 – 24; FIG. 9, block 134). Data is retrieved from the device using the device driver. (Specification, page 11, lines 24 – 25; FIG. 9, block 136).

Independent Claim 20 is directed to a system for instantiating a device driver that

includes means for dynamically associating a first software component with the device driver at run-time. The first software component contains information that facilitates communication with devices of a specific device type. (Specification, page 10, lines 13 – 18; FIG. 6). The device driver manager 86, processor 72, and memory 74 of FIG. 3 provide structure for the means for dynamically associating.

Independent Claim 31 is directed to a system for collecting data from a device that includes means for receiving a request to collect data from the device. (Specification, page 11, lines 15 – 16; FIG. 9, block 132). In addition, the system includes means for dynamically associating a software component with a device driver at run-time. The software component contains information that facilitates communication with the device. (Specification, page 11, lines 21 – 24; FIG. 9, block 134). The system further includes means for retrieving data from the device using the device driver. (Specification, page 11, lines 24 – 25; FIG. 9, block 136). The access device program 84, processor 72, and memory 74 of FIG. 3 provide structure for the means for receiving. The device driver manager 86, processor 72, and memory 74 of FIG. 3 provide structure for the means for dynamically associating. The service management system 24 of FIG. 1 provides structure for the means for retrieving.

Independent Claim 39 is directed to a computer program product for instantiating a device driver in which a computer readable storage medium has computer readable program code embodied therein. (Specification, page 4, line 14 – page 5, line 3, and page 9, line 17 – page 10, line 8). The computer readable program code includes computer readable program code for dynamically associating a first software component with the device driver at run-time. The first software component contains information that facilitates communication with devices of a specific device type. (Specification, page 10, lines 13 – 18; FIG. 6).

Independent Claim 50 is directed to a computer program product for collecting data from a device in which a computer readable storage medium has computer readable program code embodied therein. (Specification, page 4, line 14 – page 5, line 3, and page 9, line 17 – page 10, line 8). The computer readable program code includes computer readable program code for receiving a request to collect data from the device. (Specification, page 11, lines 15 – 16; FIG. 9, block 132) and computer readable program code for dynamically associating a

software component with a device driver at run-time. The software component contains information that facilitates communication with the device. (Specification, page 11, lines 21 – 24; FIG. 9, block 134). The computer readable program code further includes computer readable program code for retrieving data from the device using the device driver. (Specification, page 11, lines 24 – 25; FIG. 9, block 136).

Ground of Rejection to be Reviewed on Appeal

Claims 20 - 34 stand rejected under 35 U.S.C. §101 as being directed to non-statutory subject matter. (Office Action, page 2).

Claims 1, 2, 10, 12, 13, 15, 20, 21, 29, 31, 32, 34, 39, 40, 48, 50, 51, and 53 stand rejected under 35 U.S.C. §102(e) as being anticipated by U. S. Patent Publication No. 2002/0059474 to Camara et al. (hereinafter "Camara"). (Office Action, page 3).

Claims 9, 14, 28, 33, 47, and 52 stand rejected under 35 U.S.C. §103(a) as being unpatenable over Camara in view of Martin et al. (Professional XML). (Office Action, page 6).

Claims 3 – 5, 22 – 24, and 41 – 43 stand rejected under 35 U.S.C. §103(a) as being unpatenable over Camara in view of U. S. Patent No. 6,473,824 to Kreissig et al. (Office Action, page 7).

Claims 11, 30, and 49 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Camara in view of U. S. Patent Publication No. 2005/0034029 to Ramberg et al. (Office Action, page 8).

Claims 6, 25, and 44 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Camara in view of U. S. Patent No. 6,473,824 to Kreissig et al. and further in view of Martin et al. (Professional XML). (Office Action, page 9).

Argument

I. Claims 20 - 34 are Statutory

Claims 20 - 34 stand rejected under 35 U.S.C. §101 as being directed to non-statutory subject matter. (Office Action, page 2). Specifically, the Office Action alleges that Claims

20 - 34 are directed to software per se and, therefore, do not quality as statutory subject matter under 35 U.S.C. §101. Claims 20 - 34 are system claims written in means plus function format.

The Court of Appeals for the Federal Circuit, in its *en banc* decision *In re Donaldson Co.*, 16 F.3d 1189, 29 USPQ2d 1845 (Fed. Cir. 1994), decided that a "means-or-step-plus-function" limitation should be interpreted in a manner different than patent examining practice had previously dictated. "The plain and unambiguous meaning of paragraph six is that one construing means-plus-function language in a claim must look to the specification and interpret that language in light of the corresponding structure, material, or acts described therein, and equivalents thereof, to the extent that the specification provides such disclosure." *In re Donaldson* at 1193. "The broadest reasonable interpretation that an examiner may give means-plus-function language is that statutorily mandated in paragraph six. Accordingly the PTO may not disregard the structure disclosed in the specification corresponding to such language when rendering a patentability determination." *In re Donaldson* at 1194-1195.

The CAFC decision *In re Alappat*, 33 F.3d 1526, 31 USPQ2d 1545 (Fed. Cir. 1994) requires the USPTO to interpret means-plus-function language in a claim for Section 101 purposes in the same manner as *In re Donaldson* requires the USPTO to interpret means-plus-function language in a claim for Section 102 and/or 103 purposes. In *In re Alappat*, the sole independent claim was an apparatus claim that included only means-plus-function elements. The court found that even though many of the elements recite circuital elements that perform mathematical calculations, the "claimed invention as a whole is directed to a combination of interrelated elements which combine to form a machine for converting discrete waveform data samples into anti-aliased pixel illumination intensity data to be displayed on a display means. This is not a disembodied mathematical concept which may be characterized as an 'abstract idea,' but rather a specific machine to produce a useful, concrete, and tangible result." *In re Alappat* at 1557.

The court further held that a digital computer, once programmed, becomes a special purpose computer, a machine specially configured to perform certain tasks, and as such, is statutory subject matter. The court, however, contrasted several of its earlier decisions by

noting that "given the apparent lack of any supporting structure in the specification corresponding to the claimed 'means' elements, the court reasonably concluded that the claims at issue [in those earlier decisions] were in effect nothing more than process claims in the guise of apparatus claims." *In re Alappat* at 1555.

Based on the foregoing, it is clear that for purposes of a 35 U.S.C. §101 analysis, a claim written in -means-plus-function format must be interpreted in light of the corresponding structure, material, or acts described in the Specification and equivalents thereof. In the present case, FIG. 3 illustrates software modules stored on a computer readable medium (memory 74) and can be executed by the processor 72. Moreover, the flowcharts of FIGS. 6 - 9 illustrate embodiments of the various functions recited in Claims 20 - 34. The Specification explains that the flowchart blocks may be implemented as hardware or as computer program instructions that are provided to a processor of a computer or other programmable data processing apparatus. (Specification, page 9, lines 17 -28). In light of the structure provided in the Specification that supports the means-plus-function recitations of Claims 20 - 34, Appellants respectfully request that the rejection of Claims 20 - 34 under 35 U.S.C. §101 be reversed based on the failure of the Examiner to establish that Claims 1 - 13, 37, and 40 are directed to non-statutory subject matter.

II. Introduction to 35 U.S.C. §102 /§103 Analysis

Under 35 U.S.C. § 102, "a claim is anticipated only if each and every element as set forth in the claim is found, either expressly or inherently described, in a single prior art reference." M.P.E.P. § 2131 (quoting *Verdegaal Bros. v. Union Oil Co.*, 814 F.2d 628, 631, 2 U.S.P.Q.2d 1051, 1053 (Fed. Cir. 1987)). "Anticipation under 35 U.S.C. § 102 requires the disclosure in a single piece of prior art of each and every limitation of a claimed invention." *Apple Computer Inc. v. Articulate Sys. Inc.*, 57 U.S.P.Q.2d 1057, 1061 (Fed. Cir. 2000). "The fact that a certain result or characteristic may occur or be present in the prior art is not sufficient to establish the inherency of that result or characteristic. To establish inherency, the extrinsic evidence 'must make clear that the missing descriptive matter is necessarily present in the thing described in the reference, and that it would be so recognized by persons of

ordinary skill. Inherency, however, may not be established by probabilities or possibilities. The mere fact that a certain thing may result from a given set of circumstances is not sufficient.'" M.P.E.P. § 2112 (citations omitted).

A finding of anticipation further requires that there must be no difference between the claimed invention and the disclosure of the cited reference as viewed by one of ordinary skill in the art. *See Scripps Clinic & Research Foundation v. Genentech Inc.*, 927 F.2d 1565, 1576, 18 U.S.P.Q.2d 1001, 1010 (Fed. Cir. 1991). In particular, the Court of Appeals for the Federal Circuit held that a finding of anticipation requires absolute identity for each and every element set forth in the claimed invention. *See Trintec Indus. Inc. v. Top-U.S.A. Corp.*, 63 U.S.P.Q.2d 1597 (Fed. Cir. 2002). Additionally, the cited prior art reference must be enabling, thereby placing the allegedly disclosed matter in the possession of the public. *In re Brown*, 329 F.2d 1006, 1011, 141 U.S.P.Q. 245, 249 (C.C.P.A. 1964). Thus, the prior art reference must adequately describe the claimed invention so that a person of ordinary skill in the art could make and use the invention.

A determination under §103 that an invention would have been obvious to someone of ordinary skill in the art is a conclusion of law based on fact. *Panduit Corp. v. Dennison Mfg. Co.* 810 F.2d 1593, 1 U.S.P.Q.2d 1593 (Fed. Cir. 1987), *cert. denied*, 107 S.Ct. 2187. After the involved facts are determined, the decision maker must then make the legal determination of whether the claimed invention as a whole would have been obvious to a person having ordinary skill in the art at the time the invention was unknown, and just before it was made. *Id.* at 1596. The United States Patent and Trademark Office (USPTO) has the initial burden under §103 to establish a *prima facie* case of obviousness. *In re Fine*, 837 F.2d 1071, 5 U.S.P.Q.2d 1596, 1598 (Fed. Cir. 1988).

To establish a *prima facie* case of obviousness, the prior art reference or references when combined must teach or suggest all the recitations of the claims, and there must be some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. M.P.E.P. §2143. A patent composed of several elements is not proved obvious merely by demonstrating that each of its elements was, independently, known in the

prior art. *KSR Int'l Co. v. Teleflex Inc.*, 550 U. S. 1, 15 (2007). A corollary principle is that, when the prior art teaches away from combining certain known elements, discovery of a successful means of combining them is more likely to be unobvious. *Id.* at 12. If a technique has been used to improve one device, and a person of ordinary skill in the art would recognize that it would improve similar devices in the same way, using the technique is obvious unless its actual application is beyond his or her skill. *Id.* at 13. A Court must ask whether the improvement is more than the predictable use of prior art elements according to their established functions. *Id.* at 13. When it is necessary for a Court to look at interrelated teachings of multiple patents, the Court must determine whether there was an apparent reason to combine the known elements in the fashion claimed by the patent at issue. *Id.* at 14.

Appellants respectfully submit that the pending claims are patentable over the cited references for at least the reason that the cited references do not disclose or suggest, at least, all of the recitations of the pending independent claims. The patentability of the pending claims is discussed in detail hereinafter.

A. Independent Claims 1, 12, 20, 31, 39, and 50 are Patentable

Independent Claims 1, 12, 20, 31, 39, and 50 stand rejected under 35 U.S.C. §102(e) as being anticipated by Camara.

Claim 1 is directed to a method of instantiating a device driver and includes the following recitation:

> dynamically associating a first software component with the device driver at run-time, **the first software component containing information that facilitates communication with devices of a specific device type.** (Emphasis added).

Claim 12 is directed to a method of collecting data from a device and recites, in part:

> ...
> dynamically associating a software component with a device driver at run-time, **the software component containing information that facilitates communication with the device**;
> ... (Emphasis added).

Independent Claims 20, 31, 39, and 50 include similar recitations. As indicated above, the pending independent claims describe a software component being associated with a device driver at run-time that contains information that facilitates communication with the device. Thus, the pending independent claims recite both a device driver and a software component that is associated with the device driver at run time.

The Office Action cites paragraphs 22, 32, and 34 of Camara as disclosing the recitations of the independent claims. (Office Action, pages 3 and 4). These paragraphs, however, describe a scripting driver 66 or 120 that is used to communicate with and control a hardware device, such as a scanner. Camara explains that the "[t]he scripting driver 66, the script engine 68, and the driver script 70 for a given device together serve the function of a regular device driver (e.g., the device driver 98 in FIG. 3)." (Camara, paragraph 20). It appears that the combination of the scripting driver 66, script engine 68, and driver script 70 are alleged to correspond to the device driver element recited in the independent claims. Appellants submit that Camara does not disclose or suggest the software component that is dynamically associated with the device driver at run time and facilitates communication with the device as recited in the pending independent claims.

In response to this analysis, the Office Action cites the general scripting driver 66 shown in FIG. 2 of Camara as corresponding to the device driver recited in the independent claims. The scripting driver 120 corresponds to the general scripting driver 66 for the image capture example shown in FIG. 3 of Camara. The driver script 70 shown in FIG. 2 of Camara is cited as corresponding to the software component that facilitates communication with the device and is associated with the device driver at run time. The driver script 96 corresponds to the general driver script 70 for the image capture example shown in FIG. 3 of Camara. (Office Action, page 9). Appellants respectfully submit that in sharp contrast with the recitations of the pending independent claims, the driver script 96 is not dynamically associated with the scripting driver 120 at run time. Instead, as illustrated in FIG. 3 of Camara, for example, the scripting driver 120 is permanently associated with the driver script 96. As explained in paragraph 35 of Camara, the scanner scripting driver 120 receives a request to operate a scanner and uses a script engine 122 to access the appropriate driver

script 96 for the particular scanner 94 that is connected to the system. Note that Camara describes the scripting driver 120 as a "scanner scripting driver 120" because the driver 120 is dedicated to scanner hardware devices and is permanently associated with the driver scripts 96 that are used to communicate with the various types of scanners 94 that can be connected to the system.

It appears that the Office Action may be interpreting the sentence "[t]he Scanner Scripting Driver 120 uses the script engine 122 to interpret and execute the textual instructions in the driver script 96 at run-time to operate the scanner" in paragraph 34 of Camara as suggesting that the driver script 96 is dynamically associated with the driver 120 at run-time. Appellants submit, however, that the reference to "run-time" in paragraph 34 of Camara is in relation to the script engine 122 interpreting and executing the driver script 96 at run time. That is, the driver script 96 is a script file that is written in a language that is interpreted at run-time for execution rather than a language that is compiled into machine code and executed. Camara explains that the driver script 96 is implemented as a script instead of a compilable program because "writing a script is significantly easier than developing machine-executable programs, which are also much harder to debug." (Camara, paragraph 32, last sentence). Thus, Appellants maintain that Camara fails to disclose or suggest a software component that is dynamically associated with a device driver at run time and facilitates communication with the device as recited in the pending independent claims.

For at least the foregoing reasons, Appellants submit independent Claims 1, 12, 20, 31, 39, and 50 are patentable over the cited references and that rejected dependent Claims 2 – 11, 13 – 15, 21 – 30, 32 – 34, 40 – 49, and 51 – 53 are patentable, at least, by virtue of their depending from an allowable claim. Accordingly, Appellants respectfully request that the rejection of Claims 1 – 15, 20 – 34, and 39 – 53 be reversed based on the failure of the Examiner to establish a prima facie case of anticipation under 35 U.S.C. §102 for at least these reasons.

B. Claims 9, 14, 28, 33, 47, and 52 are Patentable

Dependent Claims 9, 14, 28, 33, 47, and 52 stand rejected under 35 U.S.C. §103(a) as

being unpatentable over Camara in view of Martin et al. (Professional XML). (Office Action, page 6). Dependent Claims 9, 14, 28, 33, 47, and 52 each depend from one of the independent Claims 1, 12, 20, 31, 39, and 50 which Appellants submit are patentable for at least the reasons discussed above in Section IIA. Appellants submit that dependent Claims 9, 14, 28, 33, 47, and 52 are patentable over the cited references at least by virtue of their depending an allowable claim. *Ex parte Ligh*, 159 U.S.P.Q. (BNA) 61, 62 (Bd. App. 1967). Accordingly, Appellants respectfully request that the rejection of Claims 9, 14, 28, 33, 47, and 52 be reversed based on the failure of the Examiner to establish a prima facie case of obviousness under 35 U.S.C. §103 for at least these reasons.

C. Claims 3 – 5, 22 – 24, and 41 – 43 are Patentable

Dependent Claims 3 – 5, 22 – 24, and 41 – 43 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Camara in view of U. S. Patent No. 6,473,824 to Kreissig et al. (Office Action, page 7). Dependent Claims 3 – 5, 22 – 24, and 41 – 43 each depend from one of the independent Claims 1, 20, and 39, which Appellants submit are patentable for at least the reasons discussed above in Section IIA. Appellants submit that dependent Claims 3 – 5, 22 – 24, and 41 – 43 are patentable over the cited references at least by virtue of their depending an allowable claim. *Ex parte Ligh*, 159 U.S.P.Q. (BNA) 61, 62 (Bd. App. 1967). Accordingly, Appellants respectfully request that the rejection of Claims 3 – 5, 22 – 24, and 41 – 43 be reversed based on the failure of the Examiner to establish a prima facie case of obviousness under 35 U.S.C. §103 for at least these reasons.

D. Claims 11. 30, and 49 are Patentable

Dependent Claims 11, 30, and 49 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Camara in view of U. S. Patent Publication No. 2005/0034029 to Ramberg et al. (Office Action, page 8). Dependent Claims 11, 30, and 49 each depend from one of the independent Claims 1, 20, and 39, which Appellants submit are patentable for at least the reasons discussed above in Section IIA. Appellants submit that dependent Claims 11, 30, and 49 are patentable over the cited references at least by virtue of their depending an allowable

claim. *Ex parte Ligh*, 159 U.S.P.Q. (BNA) 61, 62 (Bd. App. 1967). Accordingly, Appellants respectfully request that the rejection of Claims 11, 30, and 49 be reversed based on the failure of the Examiner to establish a prima facie case of obviousness under 35 U.S.C. §103 for at least these reasons.

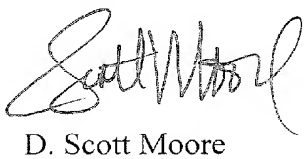
E. Claims 6, 25, and 44 are Patentable

Dependent Claims 6, 25, and 44 stand rejected under 35 U.S.C. §103(a) as being unpatentable over Camara in view of U. S. Patent No. 6,473,824 to Kreissig et al. and further in view of Martin et al. (Professional XML). (Office Action, page 9). Dependent Claims 6, 25, and 44 each depend from one of the independent Claims 1, 20, and 39, which Appellants submit are patentable for at least the reasons discussed above in Section IIA. Appellants submit that dependent Claims 6, 25, and 44 are patentable over the cited references at least by virtue of their depending an allowable claim. *Ex parte Ligh*, 159 U.S.P.Q. (BNA) 61, 62 (Bd. App. 1967). Accordingly, Appellants respectfully request that the rejection of Claims 6, 25, and 44 be reversed based on the failure of the Examiner to establish a prima facie case of obviousness under 35 U.S.C. §103 for at least these reasons.

III. Conclusion

In summary, Appellants respectfully submit that the independent Claims 1, 12, 20, 31, 39, and 50 are patentable over the cited references for at least the reason that the cited reference fails to disclose or suggest all of the recitations of each of these claims. Accordingly, Appellants respectfully request reversal of the rejection of independent Claims 1, 12, 20, 31, 39, and 50 and all pending claims depending therefrom. Appellants further submit that Claims 20 - 34 satisfy the requirements of 35 U.S.C. §101 and respectfully request reversal of the rejection of Claims 20 - 34 on this basis.

Respectfully submitted,



D. Scott Moore
Registration No. 42,011

Customer No. 20792
Myers Bigel Sibley & Sajovec
P. O. Box 37428
Raleigh, North Carolina 27627
Telephone: (919) 854-1400
Facsimile: (919) 854-1401

CERTIFICATION OF TRANSMISSION

I hereby certify that this correspondence is being transmitted via the Office electronic filing system in accordance with § 1.6(a)(4) to the U.S. Patent and Trademark Office on October 30, 2007.

Susan E. Freedman

Susan E. Freedman
Date of Signature: October 30, 2007

APPENDIX A

1. (Previously presented) A computer implemented method of instantiating a device driver, comprising:

dynamically associating a first software component with the device driver at run-time, the first software component containing information that facilitates communication with devices of a specific device type.

2. (Original) A method as recited in Claim 1, further comprising:

defining a plurality of device parameters;

associating at least one of the plurality of device parameters with a service; and

communicating the at least one of the plurality of device parameters associated with the service to the device driver.

3. (Original) A method as recited in Claim 2, wherein defining the plurality of device parameters comprises:

declaring a parameter base class that defines the plurality of device parameters;

wherein associating the at least one of the plurality of device parameters with the service comprises:

deriving a service-specific sub-class from the base class that defines the at least one of the plurality of device parameters that are associated with the service;

wherein the method further comprises:

instantiating the service-specific sub-class to create a service-specific sub-class object; and

instantiating the parameter base class to create a parameter base class object.

4. (Original) A method as recited in Claim 3, wherein communicating the at least one of the plurality of device parameters associated with the service to the device driver

comprises:

passing the at least one of the plurality of device parameters associated with the service from the service-specific sub-class object to the device driver.

5. (Original) A method as recited in Claim 1, further comprising:

defining a plurality of common device parameters;

defining a plurality of service-specific device parameters;

associating the common device parameters with the service-specific device parameters; and

communicating the common device parameters and the service-specific device parameters to the device driver.

6. (Original) A method as recited in Claim 5, wherein defining the plurality of common device parameters comprises:

declaring a parameter base class that defines the plurality of common device parameters;

wherein defining the plurality of service-specific device parameters comprises:

providing a second software component that comprises one of a script file and an extensible markup language (XML) file; and

wherein the method further comprises:

instantiating the parameter base class to create a parameter base class object.

7. (Original) A method as recited in Claim 6, wherein associating the common device parameters with the service-specific device parameters comprises:

dynamically loading the parameter base class object with the second software component at run time.

8. (Original) A method as recited in Claim 7, wherein communicating the common device parameters and the service-specific device parameters to the device driver

comprises:

passing the common device parameters and the service-specific device parameters from the parameter base class object to the device driver after loading the parameter base class object with the second software component at run time.

9. (Original) A method as recited in Claim 1, wherein the first software component comprises one of a script file and an extensible markup language (XML) file.

10. (Original) A method as recited in Claim 1, wherein dynamically associating the first software component with the device driver at run-time comprises:

selecting the first software component from a plurality of software components, respective ones of the plurality of software components being associated with respective ones of a plurality of device types.

11. (Original) A method as recited in Claim 10, further comprising:

generating the plurality of software components based on a plurality of management information base (MIB) files, respective ones of the plurality of MIB files being associated with respective ones of the plurality of device types.

12. (Previously presented) A computer implemented method of collecting data from a device, comprising:

receiving a request to collect data from the device;

dynamically associating a software component with a device driver at run-time, the software component containing information that facilitates communication with the device; and

retrieving data from the device using the device driver.

13. (Original) A method as recited in Claim 12, wherein retrieving data from the device using the device driver comprises:

associating at least one device parameter with a service;

communicating the at least one device parameter to the device driver; and

retrieving data associated with the at least one device parameter from the device.

14. (Original) A method as recited in Claim 12, wherein the first software component comprises one of a script file and an extensible markup language (XML) file.

15. (Original) A method as recited in Claim 12, wherein dynamically associating the software component with the device driver at run-time comprises:

selecting the first software component from a plurality of software components, respective ones of the plurality of software components being associated with respective ones of a plurality of device types.

20. (Original) A system for instantiating a device driver, comprising:

means for dynamically associating a first software component with the device driver at run-time, the first software component containing information that facilitates communication with devices of a specific device type.

21. (Original) A system as recited in Claim 20, further comprising:

means for defining a plurality of device parameters;

means for associating at least one of the plurality of device parameters with a service; and

means for communicating the at least one of the plurality of device parameters associated with the service to the device driver.

22. (Original) A system as recited in Claim 21, wherein the means for defining the plurality of device parameters comprises:

means for declaring a parameter base class that defines the plurality of device parameters;

wherein the means for associating the at least one of the plurality of device parameters with the service comprises:

means for deriving a service-specific sub-class from the base class that defines the at least one of the plurality of device parameters that are associated with the service;

wherein the system further comprises:

means for instantiating the service-specific sub-class to create a service-specific sub-class object; and

means for instantiating the parameter base class to create a parameter base class object.

23. (Original) A system as recited in Claim 22, wherein the means for communicating the at least one of the plurality of device parameters associated with the service to the device driver comprises:

means for passing the at least one of the plurality of device parameters associated with the service from the service-specific sub-class object to the device driver.

24. (Original) A system as recited in Claim 20, further comprising:

means for defining a plurality of common device parameters;

means for defining a plurality of service-specific device parameters;

means for associating the common device parameters with the service-specific device parameters; and

means for communicating the common device parameters and the service-specific device parameters to the device driver.

25. (Original) A system as recited in Claim 24, wherein the means for defining the plurality of common device parameters comprises:

means for declaring a parameter base class that defines the plurality of common device parameters;

wherein the means for defining the plurality of service-specific device parameters

comprises:

means for providing a second software component that comprises one of a script file and an extensible markup language (XML) file; and

wherein the system further comprises:

means for instantiating the parameter base class to create a parameter base class object.

26. (Original) A system as recited in Claim 25, wherein the means for associating the common device parameters with the service-specific device parameters comprises:

means for dynamically loading the parameter base class object with the second software component at run time.

27. (Original) A system as recited in Claim 26, wherein the means for communicating the common device parameters and the service-specific device parameters to the device driver comprises:

means for passing the common device parameters and the service-specific device parameters from the parameter base class object to the device driver after loading the parameter base class object with the second software component at run time.

28. (Original) A system as recited in Claim 20, wherein the first software component comprises one of a script file and an extensible markup language (XML) file.

29. (Original) A system as recited in Claim 20, wherein the means for dynamically associating the first software component with the device driver at run-time comprises:

means for selecting the first software component from a plurality of software components, respective ones of the plurality of software components being associated with respective ones of a plurality of device types.

30. (Original) A system as recited in Claim 29, further comprising:

means for generating the plurality of software components based on a plurality of management information base (MIB) files, respective ones of the plurality of MIB files being associated with respective ones of the plurality of device types.

31. (Original) A system for collecting data from a device, comprising:

means for receiving a request to collect data from the device;

means for dynamically associating a software component with a device driver at run-time, the software component containing information that facilitates communication with the device; and

means for retrieving data from the device using the device driver.

32. (Original) A system as recited in Claim 31, wherein the means for retrieving data from the device using the device driver comprises:

means for associating at least one device parameter with a service;

means for communicating the at least one device parameter to the device driver; and

means for retrieving data associated with the at least one device parameter from the device.

33. (Original) A system as recited in Claim 31, wherein the first software component comprises one of a script file and an extensible markup language (XML) file.

34. (Original) A system as recited in Claim 31, wherein the means for dynamically associating the software component with the device driver at run-time comprises:

means for selecting the first software component from a plurality of software components, respective ones of the plurality of software components being associated with respective ones of a plurality of device types.

39. (Original) A computer program product for instantiating a device driver, comprising:

a computer readable storage medium having computer readable program code embodied therein, the computer readable program code comprising:

computer readable program code for dynamically associating a first software component with the device driver at run-time, the first software component containing information that facilitates communication with devices of a specific device type.

40. (Original) A computer program product as recited in Claim 39, further comprising:

computer readable program code for defining a plurality of device parameters;

computer readable program code for associating at least one of the plurality of device parameters with a service; and

computer readable program code for communicating the at least one of the plurality of device parameters associated with the service to the device driver.

41. (Original) A computer program product as recited in Claim 40, wherein the computer readable program code for defining the plurality of device parameters comprises:

computer readable program code for declaring a parameter base class that defines the plurality of device parameters;

wherein the computer readable program code for associating the at least one of the plurality of device parameters with the service comprises:

computer readable program code for deriving a service-specific sub-class from the base class that defines the at least one of the plurality of device parameters that are associated with the service;

wherein the computer program product further comprises:

computer readable program code for instantiating the service-specific sub-class to create a service-specific sub-class object; and

computer readable program code for instantiating the parameter base class to create a parameter base class object.

42. (Original) A computer program product as recited in Claim 41, wherein the computer readable program code for communicating the at least one of the plurality of device parameters associated with the service to the device driver comprises:

computer readable program code for passing the at least one of the plurality of device parameters associated with the service from the service-specific sub-class object to the device driver.

43. (Original) A computer program product as recited in Claim 39, further comprising:

computer readable program code for defining a plurality of common device parameters;

computer readable program code for defining a plurality of service-specific device parameters;

computer readable program code for associating the common device parameters with the service-specific device parameters; and

computer readable program code for communicating the common device parameters and the service-specific device parameters to the device driver.

44. (Original) A computer program product as recited in Claim 43, wherein the computer readable program code for defining the plurality of common device parameters comprises:

computer readable program code for declaring a parameter base class that defines the plurality of common device parameters;

wherein the computer readable program code for defining the plurality of service-specific device parameters comprises:

computer readable program code for providing a second software component that comprises one of a script file and an extensible markup language (XML) file; and

wherein the computer program product further comprises:

computer readable program code for instantiating the parameter base class to create a

parameter base class object.

45. (Original) A computer program product as recited in Claim 44, wherein the computer readable program code for associating the common device parameters with the service-specific device parameters comprises:

computer readable program code for dynamically loading the parameter base class object with the second software component at run time.

46. (Original) A computer program product as recited in Claim 45, wherein the computer readable program code for communicating the common device parameters and the service-specific device parameters to the device driver comprises:

computer readable program code for passing the common device parameters and the service-specific device parameters from the parameter base class object to the device driver after loading the parameter base class object with the second software component at run time.

47. (Original) A computer program product as recited in Claim 39, wherein the first software component comprises one of a script file and an extensible markup language (XML) file.

48. (Original) A computer program product as recited in Claim 39, wherein the computer readable program code for dynamically associating the first software component with the device driver at run-time comprises:

computer readable program code for selecting the first software component from a plurality of software components, respective ones of the plurality of software components being associated with respective ones of a plurality of device types.

49. (Original) A computer program product as recited in Claim 48, further comprising:

computer readable program code for generating the plurality of software components

based on a plurality of management information base (MIB) files, respective ones of the plurality of MIB files being associated with respective ones of the plurality of device types.

50. (Original) A computer program product for collecting data from a device, comprising:

a computer readable storage medium having computer readable program code embodied therein, the computer readable program code comprising:

computer readable program code for receiving a request to collect data from the device;

computer readable program code for dynamically associating a software component with a device driver at run-time, the software component containing information that facilitates communication with the device; and

computer readable program code for retrieving data from the device using the device driver.

51. (Original) A computer program product as recited in Claim 50, wherein the computer readable program code for retrieving data from the device using the device driver comprises:

computer readable program code for associating at least one device parameter with a service;

computer readable program code for communicating the at least one device parameter to the device driver; and

computer readable program code for retrieving data associated with the at least one device parameter from the device.

52. (Original) A computer program product as recited in Claim 50, wherein the first software component comprises one of a script file and an extensible markup language (XML) file.

53. (Original) A computer program product as recited in Claim 50, wherein the computer readable program code for dynamically associating the software component with the device driver at run-time comprises:

computer readable program code for selecting the first software component from a plurality of software components, respective ones of the plurality of software components being associated with respective ones of a plurality of device types.

APPENDIX B – EVIDENCE APPENDIX

None

APPENDIX C – RELATED PROCEEDINGS APPENDIX

None.